Robotic Collision Avoidance Simulation Using a Two-Wheeled Robot and LiDAR Sensor

Mary Stirling Brown

Samford University  
800 Lakeshore Dr  
Birmingham, AL 35229  
+1 251 458 5466

mbrown35@samford.edu

Brian Toone

Samford University  
800 Lakeshore Dr  
Birmingham, AL 35229  
+1 205 726 2960

brtoone@samford.edu

**ABSTRACT**

Autonomous vehicles typically use a variety of sensors, including LiDAR sensors, to navigate through an environment and avoid obstacles without intervention. A LiDAR works by measuring the time it takes for the radiated laser to be reflected off an object back to the sensor in order to obtain a distance measurement to objects in the environment. In this research, a two-wheeled robot with a single-layer, rotating LiDAR sensor is used to test a collision avoidance algorithm in a simulated environment using Webots software. The collision avoidance algorithm is based on algorithms used in previous research. The created algorithm analyzes the LiDAR data to turn a specific angle away from the object to reach its target destination without collision. The robot has GPS and compass to aid in navigation towards its target destination. An environment with various obstacles is set up to test the algorithms. The proposed algorithm was unable to be tested due to a software bug and time limitations.

**Categories and Subject Descriptors**

**[Computer Systems Organization]:** Embedded and Cyber-physical systems – *robotics, robotic autonomy, sensors and actuators.*

**[Theory of computation]**: Data structures and algorithms for data management.

**General Terms**

Algorithms, Design, Experimentation

# INTRODUCTION

Autonomous robotics has spread to many different sectors of people’s everyday lives, from autonomous cars, industrial manufacturing, food deliveries, and land mapping and surveying. Collision avoidance algorithms is a current research topic in Computer Science to enable robots to autonomously navigate. Various sensors, such as sonar, light sensors, cameras, and many more, have been used in this research. Another viable option is a LiDAR sensor, standing for Light Detection and Ranging. The LiDAR shoots pulses of laser light to find distance and other information from the target. It calculates the distance from an object by measuring the time a laser is detected from the sensor to being reflected to the sensor. The standard way to store this LiDAR data is with polar coordinates with the angle and distance measurement. LiDAR data is typically used to generate 2-D or 3-D point clouds of the scanned environment [3].

The next section of this paper details the background with a literature review. Section 3 focuses on the methodology and algorithms that are implemented in this research for a successful collision avoidance system. This procedure is tested in a simulated environment, and its results are presented in Section 4. Finally, Section 5 talks about the conclusions of this research as well as limitations and future work.

# BACKGROUND

According to Hutabarat, *et al.*, the LiDAR technology proved to be successful in that the robot was able to avoid objects of certain colors in its navigation. It does this by using the Braitenburg algorithm where sensor input affects the motor speed. This research concluded that the color of an object and the intensity of ambient light do not affect the distance measurements taken by the LiDAR [3].

In research by Baras, *et al.*, a robot is successfully able to navigate in an unknown environment by only using a single LiDAR in their robotic design with no other sensors. The robot in their tests was able to avoid obstacles through the implementation of a point cloud from the LiDAR data [6].

According to Wu, *et al.*, a collision avoidance algorithm is developed based off a median filter that replaces a group of points with a midpoint. The scanned area is then grouped by emergency, accurate, and fuzzy obstacle-avoidance areas. Tests revealed that this approach was more efficient than other single collision avoidance algorithms [7].

Research by Park, *et al.*, reveals that the Artificial Potential Fields algorithm works by the robot being attracted to its waypoint, or target destination, and repelled by a negative charge, or obstacle. There are limitations to this when the robot gets trap in local minima of an environment [5].

# METHODOLOGY AND ALGORITHMS

## Hardware and Software

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*Samford University Department of Math and Computer Science*, May 2011, Birmingham, AL. Copyright 2011 Samford University…$10.00.

The simulation software used in this research is called Webots. It allows for the development of different 3-D environments with different physical factors, such as mass, gravity, friction, and many more. A concept called time-step is used to connect the virtual time to real-world physics. All this information is stored in a *world*. Robots in this world can be programmed by creating a *controller* with various programming languages. In the case of this research, Python is used to code the robot [2].

A picture containing toy

Description automatically generated**Figure 1: GCTronic’s e-puck model**

The robot in this research is the GCTronic’s e-puck, as seen in Figure 1. The e-puck is a two-wheeled robot where its movement is determined through differential driving techniques. As seen in Figure 2, it has eight infra-red (IR) sensors and eight proximity sensors surrounding the edge of the e-puck robot. The IR sensors have LED that lights up when an object is within four centimeters of the robot.

Diagram, schematic

Description automatically generated**Figure 2: sensor diagram**

The LiDAR designed in this research is fixed on the front side of the e-puck robot. It is built with the features of having a single layer of scanned distance data with a max distance range of five meters. It has a 90-degree field of view with 25 distance readings, or horizontal resolution, for one layer at a time.

A GPS is integrated on the e-puck robot as well in order to track the robot’s position and the target destination’s coordinates. A compass is also needed to be able to turn the robot toward its target destination.

## Collecting the LiDAR Distance Data

The LiDAR data scale has a field of view of 90° with 25 distance readings per scan. The angle between each distance measurement was calculated by the following equation.

A counter variable keeps track of which distance reading was happening at each data point, from 1 to 25. The overall angle of that distance measurement is added up to get the measurement from the 0° angle, which is the first distance measurement in a complete scan. Therefore, the 45° angle measurement is the distance scan measurement directly in front of the e-puck.

The LiDAR measures the distance of an object in meters. This information and the angle measurement is stored in a Python tuple where the first position is the angle measurement from 0° and the second position is the LiDAR distance from objects in meters.

## Differential Drive Procedure

The e-puck robot uses the concept of differential drive, meaning that the two wheels have an independent actuator and movement from the other wheel. This concept was used to be able to turn the robot left and right by setting the motor velocity with the appropriate speed. The following table shows what to set the velocity as for each motor depending on the movement.

Table : differential drive velocities

|  |  |  |
| --- | --- | --- |
| **Movement** | **Left motor velocity** | **Right motor velocity** |
| Forward | + | + |
| Backward | - | - |
| Right turn | + | - |
| Left turn | - | + |

The robot needs to know how to turn a specific angle. Important pieces of data from the e-puck needed to know this is that the axle length is 0.52 meters, the wheel radius is 0.0205 meters, and the angular speed is 6.28 radians.

The following method and equations allow us to turn the robot for a specified angle. First, we need to calculate the tangential speed, which is the linear speed of the robot moving in a circular rotation. The equation is used:

The tangential speed is therefore 0.12874 meters per second by using the e-puck angular speed and wheel radius. It is then calculated to see how many degrees the e-puck can move in a second by the following equation:

So, the e-puck can rotate approximately 278.237 degree per second. It can now be calculated to see how long the robot should turn in order for it to turn a specific angle:

Since the robot can turn for a specific angle, the time duration to move forward a certain distance is calculated by:

Therefore, the e-puck robot integrates these formulas with the differential drive method in Table 1 to specify the amount and direction in which to turn or move.

## Collision Avoidance Algorithm

A collision avoidance algorithm is designed for the robot to reach a target destination without running into obstacles. It was inspired by the Artificial Potential Field method in that it avoids obstacles and is attracted to the target destination in that it is periodically adjusting the heading angle to it. The collision avoidance algorithm that is developed uses the LiDAR and the eight infra-red (IR) sensors that are on the e-puck robot.

Diagram

Description automatically generated**Figure 3: LiDAR field of view for collision avoidance**

First, the LiDAR’s 90° field of view is split into two 45° angles labeled with a left and right side, as seen in Figure 3. The purpose is for the robot to know which side to turn to avoid an obstacle. In Figure 4, a collision avoidance algorithm is proposed.

Timeline

Description automatically generated**Figure 4: flowchart of the collision avoidance algorithm**

When the robot is moving in a different direction to avoid the obstacle, the robot will not adjust its heading angle until the IR sensors on the side with the obstacle are no longer lit up. This would mean that the obstacle would have been passed. This algorithm will therefore keep track of obstacles on all sides of the robot because of the IR sensors on all sides and the forward-facing LiDAR to detect obstacles in the robot’s path.

# TESTING

The collision avoidance algorithm was not able to be tested due to the Webots software crashing and time limitations. However, a simulation environment in a world in Webots was developed with various boxes for the robot to navigate around to its target destination. The white circle is the robot’s target destination on the left, and the e-puck robot is the green circular object on the right, as seen in Figure 5.

A picture containing box, different, various, wood

Description automatically generated**Figure 5: world environment for testing simulation**

# CONCLUSION

A collision avoidance algorithm was developed to mimic previous robotic research. One main result of this was that LiDAR data was able to be stored and manipulated for obstacle detection without integration of a point cloud, which takes up a lot of data and time when performing collision avoidance algorithms. Another result was implementing differential drive in Webots software to turn a certain angle with integration of LiDAR data. The created algorithm was not able to be tested due to Webots crashing and time limitations. Despite this, a simulation environment was created to test future collision avoidance algorithms.

## What you learned

The purpose of this project was to study and learn more about the field of robotics and collision avoidance algorithms. Even though it was not able to be developed and tested fully due to software bugs and time limitations, there were many important concepts that I was able to learn through this research. First, I learned how Webots connects physical elements to a virtual environment through the creation of worlds and controllers with implementation of the timestep variable in controllers. I learned how to integrate sensors, specifically a LiDAR, with a robot. I analyzed the effects of changing many of the properties of the LiDAR to get the most optimal LiDAR data for my collision avoidance purposes. Overall, I learned more about the field of robotics, while also learning more about simulation software for the first time.

## Where to go from here

Future research can include fully implementing the collision avoidance algorithm in Webots to see if it can reach its target destination in the simulated environment created. This would involve further programming a controller to avoid obstacles and integrating all the previous code for differential drive, a LiDAR sensor, and a GPS and compass system. Other future research could be to implement many different simulation environments with different types of obstacles rather than only boxes. This would lead to more conclusive results when testing the algorithm. Overall, this research this semester will help me in graduate school as I plan to research and specialize my classes in robotics, and this project gave me a more solid foundation in the field of robotics.

# ACKNOWLEDGMENTS

This research is supported by the Department of Mathematics and Computer Science at Samford University.

# REFERENCES

1. Albert Alfrianta. 2021. Simple avoidance algorithm, implemented on e-puck robot and simulated on Webots Robot simulator. (August 2021). Retrieved April 14, 2022 from https://medium.com/@albert.brucelee/3143c096d285
2. Cyberbotics. 1998. Webots: robot simulator. Retrieved from https://cyberbotics.com#cyberbotics
3. D. Hutabarat, M. Rivai, D. Purwanto and H. Hutomo, "Lidar-based Obstacle Avoidance for the Autonomous Mobile Robot," 2019 12th International Conference on Information & Communication Technology and System (ICTS), 2019, pp. 197-202, doi: 10.1109/ICTS.2019.8850952.
4. Gada, Kajal. *How to Use Lidars in Webots? // Webots Lidar Code in Python Tutorial*. *YouTube*, YouTube, 9 Sept. 2021, https://www.youtube.com/watch?v=7A9h7d-XWSA. Accessed 14 Apr. 2022.
5. Min Gyu Park, Jae Hyun Jeon and Min Cheol Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No.01TH8570), 2001, pp. 1530-1535 vol.3, doi: 10.1109/ISIE.2001.931933.
6. N. Baras, G. Nantzios, D. Ziouzios and M. Dasygenis, "Autonomous Obstacle Avoidance Vehicle Using LIDAR and an Embedded System," 2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST), 2019, pp. 1-4, doi: 10.1109/MOCAST.2019.8742065.Fröhlich, B. and Plate, J. The cubic mouse: a new device for three-dimensional iput. In *Proceedings of the SIGCHI conference on Human factors in computing systems   
   (CHI ’00)* (The Hague, The Netherlands, April 1-6, 2000). ACM Press, New York, NY, 2000, 526-531.
7. P. Wu, S. Xie, H. Liu, J. Luo and Q. Li, "A novel algorithm of autonomous obstacle-avoidance for mobile robot based on LIDAR data," 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2015, pp. 2377-2382, doi: 10.1109/ROBIO.2015.7419694.Sannella, M. J. *Constraint Satisfaction and Debugging for Interactive User Interfaces.* Ph.D. Thesis, University of Washington, Seattle, WA, 1994.